AFRL-VA-WP-TP-2006-312

DECISION MAKING UNDER UNCERTAINTY APPLIED TO THE COUNTER SCENARIO (PREPRINT)

Anouck Girard

JANUARY 2006

Approved for public release; distribution is unlimited.

STINFO INTERIM REPORT

AIR VEHICLES DIRECTORATE
AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

NOTICE

PAO Case Number: AFRL/WS-06-0187, 24 Jan 2006.

THIS TECHNICAL REPORT IS APPROVED FOR PUBLICATION.

/s/

Phillip Chandler
Senior Aerospace Engineer
Control Design and Analysis Branch
Air Force Research Laboratory
Air Vehicles Directorate

/s/

Deborah S. Grismer
Chief
Control Design and Analysis Branch
Air Force Research Laboratory
Air Vehicles Directorate

/s/

Brian W. Van Vliet
Chief
Control Sciences Division
Air Force Research Laboratory
Air Vehicles Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
January 2006	Conference Paper Preprint	06/03/2005– 09/30/2005

4. TITLE AND SUBTITLE
DECISION MAKING UNDER UNCERTAINTY APPLIED TO THE COUNTER SCENARIO (PREPRINT)

5a. CONTRACT NUMBER
In-house

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)
Anouck Girard

5d. PROJECT NUMBER
N/A

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Control Design and Analysis Branch (AFRL/VACA)
Control Sciences Division
Air Vehicles Directorate, Air Force Materiel Command
Air Force Research Laboratory
Wright-Patterson Air Force Base, OH 45433-7542

8. PERFORMING ORGANIZATION REPORT NUMBER
AFRL-VA-WP-TP-2006-312

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/VACA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-VA-WP-TP-2006-312

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Report contains color.

This work has been submitted to the 2006 AIAA Guidance, Navigation, and Control Conference proceedings. This is a work of the U.S. Government and is not subject to copyright protection in the United States.

PAO Case Number: AFRL/WS-06-0187 (cleared January 24, 2006).

14. ABSTRACT
A team of micro air vehicles (MAV) are inspecting a sequence of objects to accumulate evidence to convict/acquit an object as a target. Images are sent to a remote operator who serves the role as feature detector. This apart models the MAV-operator team as a stochastic dynamic program (DP). Solution of the DP yields the optimal policy of actions to perform on the objects regardless of the operator evaluation or when it is received.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
Unclassified	Unclassified	Unclassified	SAR	40	Phillip Chandler
					19b. TELEPHONE NUMBER *(Include Area Code)* (937) 255-8680

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

DECISION MAKING UNDER UNCERTAINTY

APPLIED TO THE COUNTER SCENARIO

The report describes joint work with **Phil Chandler** of AFRL/VACA and **Meier Pachter** of AFIT. The authors would also like to thank Steve Rasmussen and Mike Patzek, both of AFRL, for constructive discussions.

This work has been submitted for publication in the proceedings of the 2006 AIAA Guidance, Navigation, and Control Conference (to be held 24 Aug 06 in Keystone, CO), AIAA publisher.

INTRODUCTION

We consider a simplified scenario for intelligence gathering in a urban environment using a Small Air Vehicle (or SAV, with a wingspan of roughly 6 feet) and several Micro Air Vehicles (or MAV, with a wingspan on the order of 12 inches). We abstract the urban environment as a grid. Vehicles are parked on the streets (lines of the grid) at arbitrary angles. Regular (or "clutter") vehicles outnumber vehicles of interest by a ratio of 10 to 1. A Small Air Vehicle is dispatched to fly over the area. It carries up to 4 MAV that it can release to gather information about a small number of vehicles. The SAV can determine vehicle positions accurately, but not orientations. It is equipped with a sensor that enables it to select vehicles for examination by MAVs. This sensor has a confusion matrix, which is known ahead of time.

Once dispatched by the SAV, a MAV (Micro Air Vehicle) has to fly over N vehicles for classification purposes. The list of vehicles is provided by a planner onboard the SAV and the sequence of vehicles is fixed. A certain fraction of the vehicles are known to be of interest. The MAV flies over each vehicle, takes a reading (for example, a picture) and transmits the reading to a human operator for classification. The MAV flies towards its next vehicle as it waits for an answer from the human operator.

The human operator is classifying the images by looking for a feature, F. Vehicles of interest carry this feature, which is only visible from a 90 degree range of aspect angles and a 20 degree depression angle. The human operator is not a perfect classifier: it takes a delay to get the answer, and the operator is characterized by a confusion matrix. In addition, the human operator's workload can get saturated if he receives more than four pictures a minute to classify.

After some delay, the operator answers with a classification of either "OF" (operator sees feature) or "ONF" (operator does not see feature). "OF" indicates that the vehicle is of interest with probability $p(T|OF) > P(T|ONF)$, that is, "ONF" indicates more ambiguity about the vehicle.

When the answer from the operator is received, the MAV has the option to either continue on to the next target, or turn around and go take a 2^{nd} look at the vehicle. If the MAV takes a 2^{nd} look, it will get another reading (for example, ("OF, OF") or ("OF, ONF")). The cost of taking a second look includes a fixed cost to turn around (the cost of changing direction by 180 degrees, twice), plus the delay caused by having to travel back to the initial vehicle again, and back. The MAV has limited flight time, M, which includes a short fuel reserve.

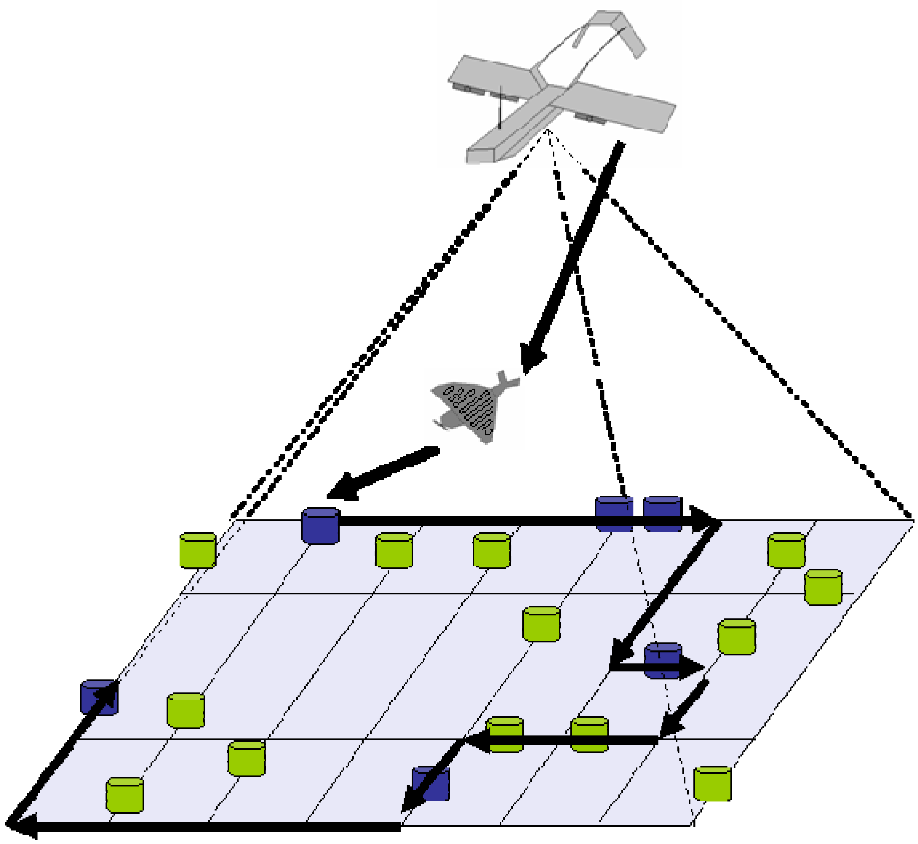

Figure 1. Cooperative Operations in UrbaN TERrain (COUNTER) abstracted scenario. The grid represents city streets. The cylinders represent vehicles on the city streets. The SAV is flying high above the region, carrying the MAV under its wings. It selects some vehicles (the blue cylinders) for MAV1 to visit, plans a route, downloads it to MAV1, and releases MAV1. MAV1 flies its route, but has a small fuel reserve, which it can use to revisit a small number of vehicles.

Sources of error in this problem are as shown in figure 2. We have the following information about the problem: original ratio of interest vehicles to clutter vehicles, SAV confusion matrix, statistical characterization of transmission and operator delays, operator confusion matrix, information about operator workload and saturation, MAV flight time. Our goal is to collect the most useful information about the vehicles. When should the MAV take second looks?

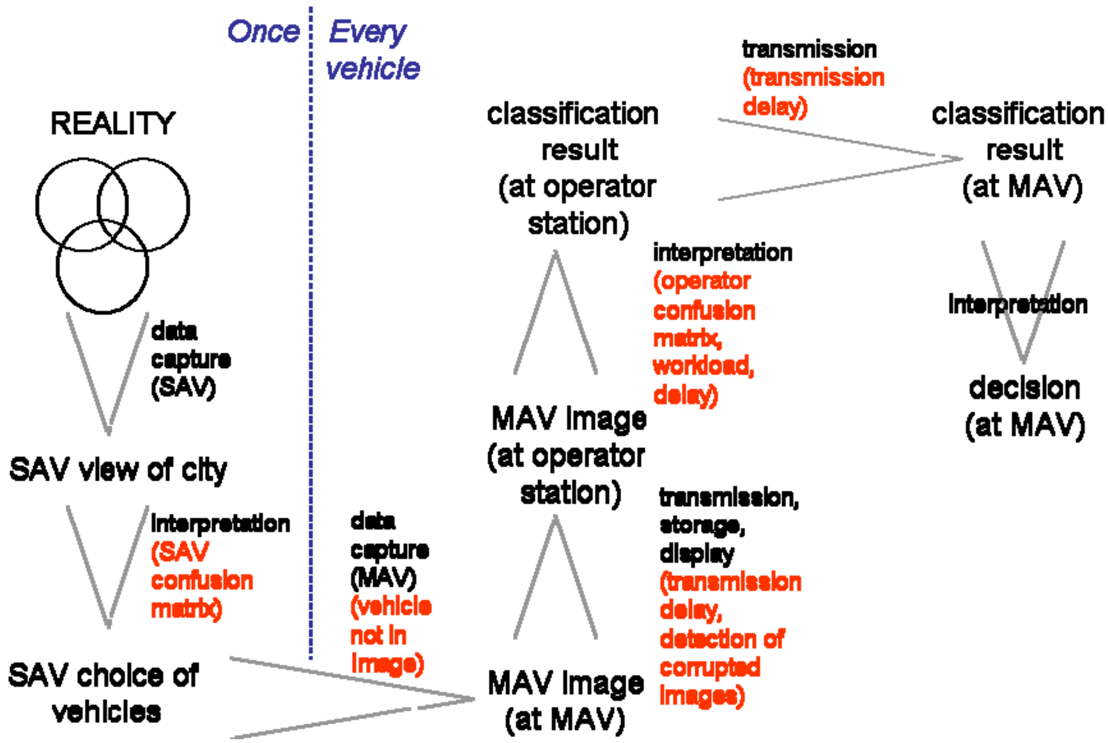

Figure 2. Sources of error in COUNTER abstracted scenario.

PROBLEM FORMULATION

We consider first the case of a perfect operator. That is, if the feature is present, the operator will find it every time, and if the feature is not present, the operator will never declare that he/she has seen it (no false positives). We will treat the case of operator confusion further in the paper.

We first discuss phrasing the problem as a problem of dynamic programming or routing under multiple constraints. We then present the approach that we decided to adopt, stochastic dynamic programming applied to sequential stochastic allocation.

DYNAMIC PROGRAMMING FORMULATION

The state of the system is given by: $s = (x_k, w_k, r_k, s_v)$.

x_k = distance traveled (assume constant speed)

w_k = measurement

r_k = operator delay. We may know the distribution for rk, or just bounds.

For now, we assume that the operator delay is less than the travel time to the next target.

s_v = set of sites visited

k indicates decision points (not fixed intervals of time).

S is the set of allowable states.
"A_s" is the set of allowable actions in state $s \in S$. $A_s = \{\text{continue, 2look}\}$.
Continue is the default action. In some states it is possible to take the 2nd look action "2look".
Taking action a in state s incurs the cost ct(s, a).
Constraint: x(final) <= x_max = speed V(cst) * M

Approach: over a finite horizon (can be adjusted), minimize the expected value of the part of $V_k(s)$ that has yet to be determined.

$$V_k(s) = \min_{a \in As} E\left[\sum_{k=T}^{pT} c_k(s,a)\right]$$

Bellman recursion:

$$V_k(s) = \min_{a \in As}\left[c_k(s,a) + \sum_j P^a(s,j)V_{k+1}(j)\right]$$

Here P^a is the transition matrix.

Obtaining an expression for costs:

A first pass expression might be a convex combination of two costs, in an expression such as:

$$c_k(s,a) = \alpha\left[1 - benefit(s,a)\right] + (1-\alpha).delay(s,a)$$

Benefit(s,a) could be a measure of entropy or another measure such as abs(p(T)now-p(T)before).

Delay(s,a) is a simpler measure and reflects the time delays inherent in the actions. The cost of taking a second look includes a fixed cost to turn around (the cost of changing direction by 180 degrees, twice), plus the cost of having to travel back to the first target again, and back. This in turn depends on the operator delay, and how far along one is in the current leg of the trip when one gets the classification result back from the expert human operator. That is, the cost is the fixed cost of a U-turn, plus the time it takes to go back to the first target, and then get back to where one was.

Using a convex combination allows us to combine benefits and delays into a single measure (basically, to turn a multi-criteria optimization problem into a single criteria one). The problem with doing this is that depending on the choice of alpha, some good solutions might be missed.

OTHER POSSIBLE OPTIMIZATION CRITERIA

It is possible to consider other optimization criteria for this problem. This method can be used as long as the criterion on the reward (as stated above) is satisfied. Other objective functions may be amenable to stochastic dynamic programming.

Other possible optimizations for this problem include:

A *myopic optimization strategy* (as proposed by Pachter):

The initial probability distribution is a constant, that is:

$$P_i^{(0)}(T) = \frac{1}{1+\alpha} \quad \text{for i = 1, …, N}$$

At the final time we have:

$$y_M = \left[P_i^{(M)}(T) - \frac{1}{N}\sum_{i=1}^{N} P_i^{(M)}(T) \right]^2$$, which is a quantity we seek to maximize.

(This is related to the entropy approach proposed below for the sequential stochastic allocation problem).

The myopic optimization approach works as follows: at time k I can either inspect the next object, O_{i+1}, or I can revisit the current object O_i. At time k I know P_i and P_{i+1} and as a result of my action either P_i or P_{i+1} will change.

The goal is to increase y_k as much as possible, that is, maximize ($y_{k+1} - y_y$).

Note that $P_{i+1}^{(k)}(T) = \frac{1}{1+\alpha}$

Another objective function (as proposed by Chandler) is a function of:

1. the expected value of the tour
2. the total transmission and operator delay
3. a fixed time cost to take a 2nd look
4. the result of the 1st pass classification
5. the time to get to the next location
6. a penalty for false classification

Maximize [(sum of target classifications) – penalty for false classification]/sum of delays

Example: going from site i to site j in the simplified COUNTER scenario.

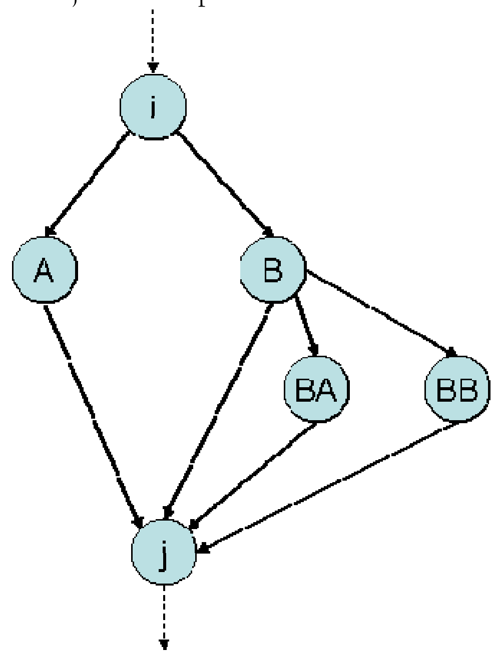

Figure 3. Graph from site i to site j.

(s,a)	Benefit (s,a)	Delay (s,a)	i	A	B	BA	BB	j
(i,cont)	0	rV	0	p(A)	p(B)	0	0	0
(A, cont)	p(T|A)-default	dij-rV	0	0	0	0	0	1
(B, cont)	p(T|B)-default	dij-rV	0	0	0	0	0	1
(B, 2look)	p(T|B)-default	rV+fixed turn cost	0	0	0	P(BA)	P(BB)	0
(BA, cont)	p(T|BA)-Previous	dij	0	0	0	0	0	1
(BB, cont)	p(T|BB)-Previous	dij	0	0	0	0	0	1

Table 1. Benefits and costs from site i to site j.

This is an example benefit (not a true measure of entropy). In the delays, r is the random operator delay, V is the fixed speed of the aircraft, dij is the distance between sites i and j. We are assuming the MAV gets an answer about site i before it reaches site j. Considering the alternative is only interesting is one can choose which site to visit next. Here the sequence is fixed. This is not meant to be an optimal decomposition into as few states as possible, it is just for the sake of the example and to develop a state transition matrix.

Interesting twists on the problem:

a. The order of sites to visit might not fixed
b. The adversary sees you fly over the first target and radios the information to all others, so that they have time to take some action (for example, camouflage or hide). What to do then? In particular, it might be useful if one can re-plan one's route here.
c. The problem can be treated in two different ways, that is, pick alpha once and minimize expected value of costs, or the operator might want to select alpha himself/herself as he/she gets more information about the search space.

ROUTING UNDER MULTIPLE CONSTRAINTS

Technically, this is not a problem of routing, but the representation can be useful for understanding the problem, especially in the framework of multiple criteria optimization, and computationally efficient algorithms exist for some of these types of problems.

Consider a directed graph G with vertices V and edges E. G=(V,E).
An edge is represented by e = (v,w,c,d), v and w are the source and destination, c is the cost (1-benefit), d is the delay.

There are paths in the graph, for example:

$$\text{Path: } v_1 \xrightarrow{c1,d1} v_2 \xrightarrow{c2,d2} \ldots \xrightarrow{cn,dn} v_{n+1}$$

Set a cost constraint C, a delay constraint D.

A path is feasible iff cost(path) <= C and delay(path) <= D.

Standard routing problem: Given the graph G=(V,E), a source node s in V, a destination t in V, find a feasible path from s to t or decide no such path exists.

Problem is NP-hard, near-polynomial solutions can be found (O(|V||E|min{C,D})).

Variation: Among feasible paths, find the one that has the lowest cost.

Efficient routing algorithms exist to prune out redundant or infeasible solutions and alleviate computational costs.

Graph will look like the one shown above repeated for the number of targets. To compute costs, a portion of costs will be deterministic, and a portion will be an expected value.

Note: there are two cases in which the problem is simple:

cost_max = max{c|(*,*,c,*)∈E}, delay_max = max{c|(*,*,*,d)∈E}

OTHER INTERESTING TWISTS/NOTES

1. The presence of the "fixed cost" for turning around and going back to look at a potential target suggests that there might be an opportunity to take advantage of "price discounts" by waiting for a few targets to build up, then go back to look at the targets all at once, if the fixed cost to U-turn is big enough.
2. The targets may not be equidistant. The distance to the next target should be a part of the decision making process.
3. It might be possible to estimate the state of the operator (for example, in terms of workload, if he/she received many requests in the last few minutes).
4. If the operator is particularly overloaded, it might be possible to adjust the speed of the aircraft (or fly holding patterns) for some fixed amount of time before moving on to the next target (eg if I don't hear anything in the next 30 seconds, I will keep going). This relates to the notion of critical level.
5. To include the effect of the opponent, we will have two competing strategies. This can be set in the framework of game theory. It will be necessary to show that the game has a value.
6. Note for the DP formulation: the optimal solution might be sensitive with respect to the transition probabilities, which could be inaccurate. May need an approach for "robustness" here (a la El Ghaoui).
7. Discounting might be considered here. It implies that rewards received in further steps will be less valuable than rewards received in the current step. It might be applicable at the end of the mission when one may or may not have enough flight time left to reach the next target, or if there is a chance that one may get shot down that increases with flight time.
8. It is possible that this problem can be phrased as a variation of the "optimal stopping" problem, in the sense that it boils down to determining a boundary between the "continue" region and the "2look" region.

SYSTEM DELAYS: STOCHASTIC SEQUENTIAL ALLOCATION

After examination, it was decided to phrase the problem as a variant of stochastic dynamic programming called stochastic sequential allocation [1], as cited in [2, 3].

Consider the following generalization of the house hunting problem, as described in [1]. Suppose there are $k \leq n$ houses to be sold. Offers arrive in a sequential manner. These offers will be assumed to be a sequence of independent, identically distributed random variables $X_1, X_2, \ldots, X_n$. The seller may accept or reject the offer but must dispose of all k houses by the n^{th} offer.

Suppose there are n cards. Let k of the cards have an associated probability equal to 1 and (n-k) of the cards have associated probability 0. If the seller accepts the j^{th} offer, he assigns it a card having an associated probability equal to 1 and receives reward x_j, and that house and card become unavailable. If the seller rejects the j^{th} offer, he assigns it a card having an associated probability equal to 0 and hence receives nothing. This procedure continues until all the houses (and cards) are disposed of. The problem is to determine which offers to accept in order to maximize the total expected profit (or reward).

The problem of deciding whether or not to take a second look with the MAV is related to the generalized house hunting problem as follows. We have a decision point whenever the first pass reading over the j^{th} site comes back from the operator after delay x_j.

There are a number of vehicles to classify, and we know the number of expected decisions points on the MAV's route. Call that quantity n. When the route is downloaded onto the MAV, its Manhattan length can be computed, and the exact time in the reserve determined. From this, a number $k \leq n$ of expected possible 2nd looks can be computed. The problem is then analogous to the house-hunting problem, that is, allocate the time in the reserve as best you can, except in our formulation, profits are a cost, the time spent on the 2nd look, and are bounded by the maximum available time for 2nd looks. In our case the expected reward for second looks is always the same – it's the time spent on the 2nd look that changes.

The key result in [1] is to show that the optimal policy is of the following form: if there are n stages to go (n cards to play), and probabilities $p_1 \leq p_2 \leq \ldots \leq p_n$, then the optimal choice in the initial stage is to use p_i (implying the i^{th} card) if the random variable X falls into the i^{th} non-overlapping interval comprising the real line. Furthermore, these intervals depend on n and the cumulative distribution function of X but are independent of the p's.

Theorem [1]: Optimal policy for sequential stochastic assignment problem
For each $n \geq 1$, there exist numbers

$$-\infty = a_{0,n} \leq a_{1,n} \leq a_{2,n} \leq \ldots \leq a_{n,n} = +\infty$$

such that whenever there are n stages to go and probabilities $p_1 \leq p_2 \leq \ldots \leq p_n$, then the optimal choice in the initial stage is to use p_i if the random variable X_1 is contained in the interval $(a_{i-1,n}, a_{i,n}]$. The $a_{i,n}$ depend on G_X, the cumulative distribution function of the random variable X, but are independent of the p's. □

This is true for a class of reward functions that have the following property. Denote by r(p,x) the expected reward if a "p" card is assigned to an "x" offer. The function r(p,x) should be differentiable and

$$\frac{\partial}{\partial x}\frac{\partial}{\partial p} r(p,x) \geq 0$$

We have discussed the form of the optimal policy, but not the calculation of the intervals $a_{i,n}$. These constants may be calculated from the result below.

Corollary [1]: Calculation of the intervals
Define $a_{0,n} = -\infty$, $a_{n,n} = +\infty$. Then,

$$a_{i,n+1} = \int_{a_{i-1,n}}^{a_{i,n}} z\,dG_X(z) + a_{i-1,n} G(a_{i-1,n}) + a_{i,n}\left[1 - G(a_{i,n})\right]$$

for i = 1, 2, …., n, and where $-\infty.0$ and $\infty.0$ are defined to be 0.

The intervals can be computed for any probability distribution of the total delays in the system (transmission delays from and to MAV, and operator delays).

NOTES ON OPERATOR MODELING

A realistic operator model for our problem might evolve over 4 different dimensions:

a. cognitive delays (can be set up to include communication delays)
b. workload (operator degrades after 2 classifications/min, effectively saturates at 4 classifications/min)
c. confusion matrix for the operator
d. degradation in image

In addition, there are two other possible answers for the operator:

a. Image was black (transmission bug, for example). Always take a second look.
b. Image doesn't have enough information to tell (site may be too dark, under foliage, in the shade (not enough contrast, etc…). Don't bother take a second look, as the second image is likely to be of the same quality.

The method presented above deals with the delays given their cumulative distribution function. The delays can be represented by any CDF.

The question now is how to include the remaining effects. We might also want to consider operator skill level (would affect the confusion matrix, and perhaps would reduce the delays, might change saturation levels), and operator to MAV ratio.

It might be possible to phrase the problem as a game where the operator plays against a number of MAVs.

It might also be possible to write a scheduler for the operator, so that the MAVs could transmit an image, a "priority" tag (1st pass classification more time sensitive than 2nd look) and a "price" (delay that the MAV is willing to wait for the answer). Images obtained on a second look are not urgent, as taking third looks yields no benefit (actually, if considering an operator confusion matrix, there are cases where it may be beneficial to take a third look).

Then the scheduler can show the operator the "most important" image at the time, and space them out to avoid overload. One thing might be that it might be worth rapidly showing the images to the operator as they come in, to get rid of the "border" cases where the image is all black (or corrupted), or where the information content is poor, then add the good ones to the queue for classification later.

BETTER CHARACTERIZATION OF OPERATOR AND TRANSMISSION DELAYS

There is no reason to expect a gamma distribution to fit actual data with extreme precision. Furthermore the fit to a histogram of reaction time data will depend on the number of trials (a single individual is unlikely to do thousands of runs) and how that data is binned. Nonetheless a shifted gamma distribution should reproduce most of the basic features that show up in a reaction time histogram: a low-end cutoff, a peak weighted toward the low end and a tail running off toward long times. The shifted gamma distribution reproduces enough of the gross features of a real reaction time distribution to be a useful model here without being unduly difficult to generate in a simulation routine.

The figure below shows the probability density function and cumulative distribution function for a distribution caused by adding three types of delays: MAV to operator delay, operator "think" time, and operator to MAV delay.

The delays were each taken to be represented by a shifted gamma distribution.

The equation for the shifted gamma distribution is a function of three parameters, as follows:

$$PDF = \frac{1}{B\Gamma(C)}\left(\frac{y-A}{B}\right)^{C-1}\exp\left(\frac{A-y}{B}\right)$$

$$CDF = \Gamma\left(C, \frac{A-y}{B}\right)$$

Where Γ(x) represents the gamma function and Γ(x, y) is the incomplete gamma function. The parameters must obey y>A, B>0 and 0<C≤100.

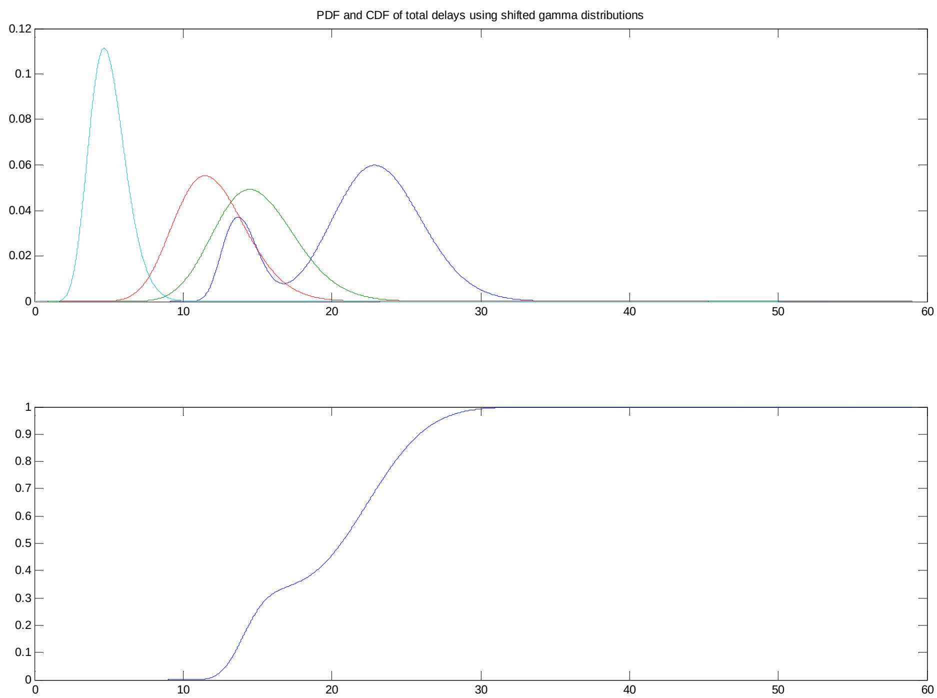

Figure 4. Possible PDF and CDF for system models using shifted gamma distributions.

The delays are added and the whole curve is shifted to the right to account for the additive nature of the minimum delays.

The mean is given by A+BC and the variance by B^2C.

MAV to Operator delays: gamma (0, 0.5, 24) (mean 12, variance 6)
Operator "think" time: gamma (0, 0.5, 30) (mean 15, variance 7.5)
Operator to MAV delays: gamma (0, 0.3, 17) (mean 5.1, variance 1.53)

Note: these numbers can be adjusted when they are known with more certainty. Keeping C as an integer is good for mathematical purposes while computing the intervals.

The critical thresholds can now be recomputed for the more accurate distribution. Note: this may involve a fair bit of mathematics, depending on the CDF. It can be done numerically.

Using shifted gamma distributions, the following formulas come in handy while computing the intervals:

$$\frac{\partial\Gamma(a,x)}{\partial x} = -x^{a-1}e^{-x}$$

$$\int x^n e^{ax}dx = \frac{x^n e^{ax}}{a} - \frac{n}{a}\int x^{n-1}e^{ax}dx = \frac{e^{ax}}{a}\left[x^n - \frac{nx^{n-1}}{a} + \frac{n(n-1)x^{n-2}}{a^2} - \ldots + \frac{(-1)^n n!}{a^n}\right]$$

Motivation for using shifted Gamma distribution comes from Ted Cohn at UCB who uses it a lot for driver and pedestrian modeling applications. Another, simpler option would be to use exponential distributions, with the advantage that it is a single-parameter skewed distribution, and it is easy to integrate/differentiate.

OPERATOR WORKLOAD

In the envisioned system, the operator is responsible for classification of images coming from up to four different MAVs. Some of the images are from first passes, some are from second passes, and the system can be configured such that if the MAV flies over a vehicle that was not selected by the SAV as part of his assigned route (as in figure 1), a picture is taken and sent to the operator anyway, as the SAV is not a perfect classifier.

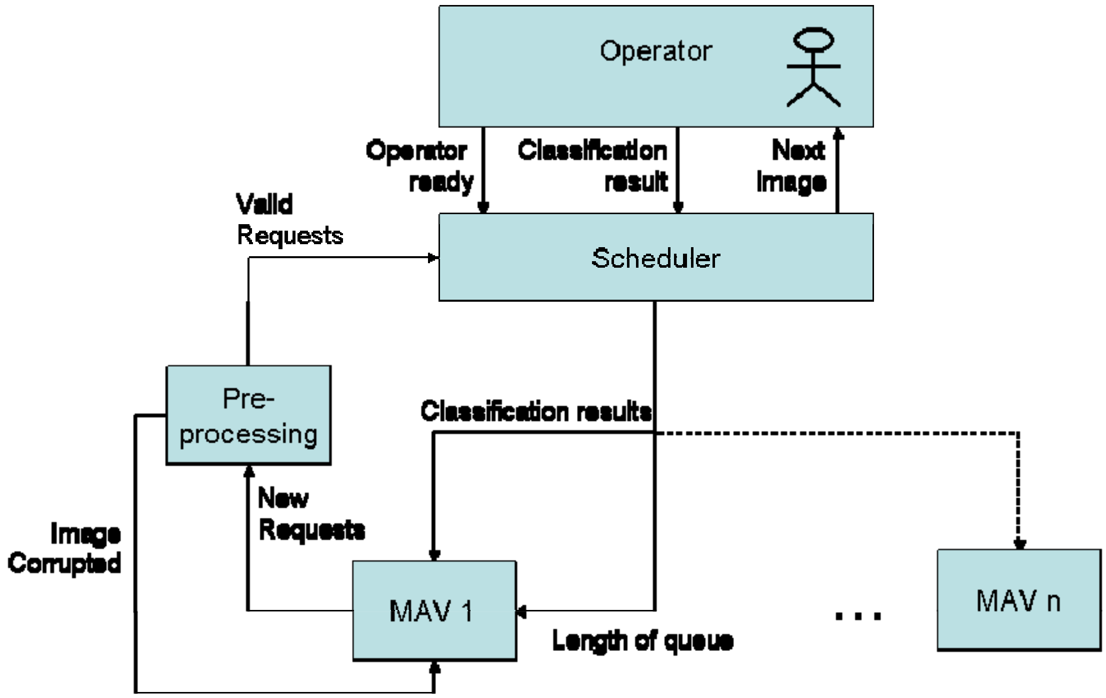

Figure 5. Operator interface to the COUNTER system and messages.

Keeping track of all the different images and priorities will presumably be a difficult task for the operator. To alleviate this, there are a number of possible strategies:

1. Pre-processing can be done on the images, as described below, to eliminate some of the corrupted images or images with poor contrast prior to showing them to the operator. A corrupted image on a first pass should yield an automatic request for a second pass, as no information was acquired. If this is done automatically delays should be small and the cost of taking a second pass minimized. An image with poor brightness/contrast is most likely unusable for classification, and should yield an automatic request to continue, as it is unlikely that a second image taken in the same conditions will yield better information.
2. The initial routes for the MAVs can be planned so that the sites to visit are spaced out in time to not saturate the operator. This is possible for very short tours, but in practice with operator delays and second looks, this will be hard to enforce.
3. The MAVs can try to estimate the state of the operator.
4. We can design a scheduler for the operator. The MAVs will send images to the operator, along with a flag indicating whether this is a 1st pass or 2nd pass image, and a "price" (delay) past which the MAVs will not be able to go back and take a second look. The scheduler, located at the operator work station, can then select which image to show the operator next. It can also send a message to the MAVs if the operator is saturated, indicating an automatic "continue" action. This might be useful if the MAVs are trying to estimate the state of the operator, as the scheduler can periodically send messages indicating the state of the "queue".

A preliminary version of a scheduler was written to interface between the MAVs and the human operator.

We chose to use Earliest Deadline First (EDF), a dynamic priority real-time scheduling policy. It is optimal, in the sense that any schedulable set of tasks can be correctly scheduled by the EDF. However, if the requests come in such that they are not schedulable (the operator saturates), then some requests will be skipped. There are many other scheduling algorithms that could be used if appropriate. For example, one could consult [4].

The scheduler's outputs are of two kinds:

1. To the operator: operator_image (request number)
2. To the MAV:
    a. The scheduler sends classification results down from the operator. This is an asynchronous message.
    b. The scheduler sends a periodic message to MAVs that have requests in the queue. This message includes:
        i. A saturated flag (yes/no)
        ii. The current length of the queue

Inputs to the scheduler include:

1. "Operator ready" message from the operator. This message is generated automatically every time the operator has classified an image, or if he has failed to return a classification before a cut-off time.
2. Requests from MAV (or from pre-processing, if applicable. Pre-processing automatically examines images for corruptions, that is large quantities of pixels that all have the same value, or for lack of contrast.).

```
while mission is ongoing
        if I got a message
                if this is a new request
                        add request to list
                        set "done" flag to 0 (pending)
                if this is an "operator ready" message
                        if target was classified
                                set that target's "done" flag to 2 (done)
                        update all remaining delays
                        if any not "done" target has a delay less than zero
                                inform MAV that classif. was not done in time
                                set done flag to 3 (skipped)
                        // figure out what image to present to operator next
                        find most urgent 1st pass request
                        if no 1st pass request found
                                find most urgent 2nd pass target
                        if no 1st or 2nd pass request founds
                                send break code (empty queue)
                        inform operator of next image
                        set next image done flag to 1 (processing)
                        calculate length of remaining queue
                        inform MAV of length of queue

        update mission time
```

Figure 6. Scheduler pseudocode.

PRE-PROCESSING: DETECTING CORRUPTED IMAGES, OR INSUFFICIENT CONTENT

It should be possible to apply standard image processing techniques to detect corrupted images before they are shown to the operator, and therefore reduce delays in the treatment of those images. Pre-processing might also improve the probability that the operator classifies images correctly.

1. Blank images can be screened out by searching for images containing either an abnormally high number of pixels with value 0 or 255, or containing "enough" successive pixels with those values.
2. The brightness and contrast can also be estimated. They can also be adjusted to improve image quality. A criterion for over-exposure can be applied.
3. It should be possible to compensate for forward motions.
4. It might be possible to estimate the amount of foliage in a picture (in case the target is under the foliage).
5. It will be much more difficult to determine whether the site to survey is actually in the frame, or in particular if the feature(s) that lead to classification is (are) in the frame. The classifier feature is considered to be in the frame if:
    a. $0 \leq$ Range ≤ 150 ft (125 ft nominal)
    b. 0 deg $\leq$ Depression (pitch) $\leq$ 45 deg (35 deg nominal)
    c. -45 deg $\leq$ Aspect (yaw) $\leq$ 45 deg (0 deg nominal)

SIMULATION RESULTS

We developed a simulation environment to test out our decision making strategies. The simulation world is a 25 by 25 Manhattan grid, which represents the streets. Each block represents 80 meters, which is consistent with the length of city blocks in Manhattan. This represents an area of approximately 2 square miles. One hundred vehicles are created on the streets, with random positions and orientations. Of these, ten (this can be varied) are vehicles of interest. This ratio of vehicles interest to clutter vehicles is assumed known ahead of time.

We simulate the SAV cueing with the following confusion matrix. After SAV cueing, we get a more favorable ratio of vehicles of interest to clutter vehicles (25%). Our reduced set contains 24 vehicles, of which 6 are vehicles of interest.

	SAV says T	SAV says NT
Vehicle of interest (T)	.6	.4
Clutter vehicle (NT)	.2	.8

Table 2. SAV confusion matrix.

The 24 vehicles are allocated to 4 different MAV for further examination. The MAV are initialized at random on the grid, and their positions and orientations are constrained to be on a street. Each MAV is allocated a tour of vehicles to visit in a fixed order. Different allocation strategies can be tested. In our example, as a preliminary solution, we are using greedy allocation. This has interesting consequences on operator workload which we will discuss later.

Once a MAV is allocated a tour, it performs path planning on a grid to reach all the vehicles in the correct order. A simple kinematical model of the vehicles is used, where the MAV are constrained to fly along city streets. The MAV are assumed to fly at constant speed (taken to be 20m/s). Costs for 90 degree turns and u-turns were estimated, where the cost of the 90 degree turn is proportional to the arc length for the turn, and the cost for a 180 degree turn in a city street is taken to be three times the cost of a 90 degree turn, as the turn would have to be three-dimensional. This variable can be adjusted, with the consequence that if the 180 degree turns become too "expensive", the MAVs fly around the block to take a second look at the vehicles instead of making 180 degree turns.

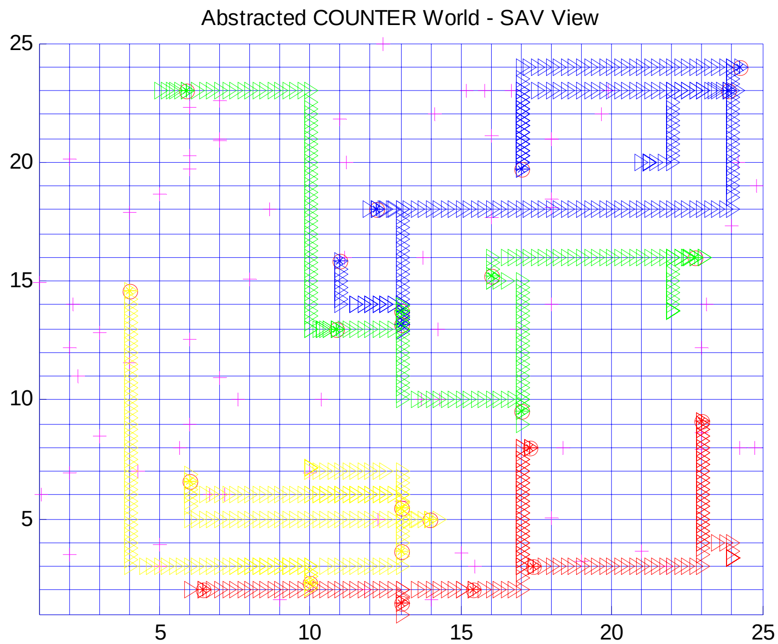

Figure 7. Example simulation run for COUNTER abstracted scenario. The test case considers an area of 25 blocks by 25 blocks (approximately 2 square miles). Streets are indicated by the blue grid. Crosses indicate 100 vehicles, initialized at random positions and orientations on the grid. 24 of those vehicles were selected according to the SAV confusion matrix for visits by the MAVs. Of those, 6 are real targets. Each (of 4) MAV flies his tour, while constrained to flying down streets, and makes decisions about which vehicles to take 2nd looks for based on operator classifications. MAV routes are shown in color.

When a MAV flies over a vehicle, it will take a picture (or sequence of snapshots) that will be sent back to the human operator for classification. At this point, our simulation is not "human-in-the-loop", and we send a command to our "operator module" indicating whether the feature was photographed ("F") or not ("NF"). The feature is only visible in a 90 degree range of aspect angles and a 20 degree depression angle, provided by the MAV. Once the message has been sent to the operator module, the default behavior for the MAv is to start flying towards the next vehicle in the tour.

The operator module returns a reading of "the operator has seen the feature ("OF" or not ("NOF") with a random delay, r, which for now is drawn from a uniform distribution between 12 and 29 seconds. This delay represents an aggregate of communications delays to/from the operator, and operator classification delays. For this distribution, the critical intervals given in section IIA can be computed. For a tour of six vehicles, the critical intervals are as follows. They can be easily recomputed for any length of tour.

i	0	1	2	3	4	5	6
$a_{i,6}$	$-\infty$	16.05	18.58	21	23.41	25.96	$+\infty$
$a_{i,5}$	$-\infty$	16.65	19.58	22.41	25.34	$+\infty$	
$a_{i,4}$	$-\infty$	17.48	21	24.51	$+\infty$		
$a_{i,3}$	$-\infty$	18.75	23.25	$+\infty$			
$a_{i,2}$	$-\infty$	21	$+\infty$				
$a_{i,1}$	$-\infty$	$+\infty$					

Table 3. Critical intervals for operator and communications delays, if delays are drawn from a uniform distribution between 12 and 29 seconds.

The decision logic for whether or not to take second looks is as follows. Second looks always give you more information about the state of the system/vehicles, as long as the operator classification is "NOF". If the operator classification is "OF", no information is gained from a second look, but no information is lost, either (flight time is lost, though). We want to collect the most information during this mission (calculating the value of this information will be addressed more formally later). In a first time we assume the operator is perfect, that is, if the feature was in the picture, the operator always recognizes it and the operator does not declare false positives. (We relax these assumptions in the next section.)

We start by computing the Manhattan length of the tour, and by including the effect of 90 and 180 degree turns. Since the vehicles are initialized randomly for every simulation run, in some cases a MAV will draw a tour for which it will have time to take 2nd looks for all vehicles within its flight time. Otherwise, it is possible to estimate conservatively what the maximum and minimum number of possible 2nd looks may be (based on cost of 180 degree turns and the distribution of delays). If the operator classification is "OF", do not take a 2nd look. If the operator classification is "NOF", then, depending on whether or not the actual delay is for each vehicle, the MAV can decide whether to take a 2nd look or not. For example, if I can take at least two 2nd looks, I will definitely take a 2nd look if (the classification is "NOF" and) the delay over a vehicle is less than 16.05, or if it is between 16.05 and 18.58. The MAV can take 2nd looks if the delay is higher, but at a risk of not finishing the tour. So, for "NOF" classifications, if the delay associated with the first vehicle is 17, the MAV will take a 2nd look, if the delay associated with the 2nd vehicle is 24, the MAV will not take a 2nd look, if the delay associated with the 3rd vehicle is 17, take a 2nd look at your own risk, if the delay associated with the 4th vehicle is 15, take a 2nd look, etc… (Delays are expressed in seconds).

In addition, the last vehicle in each tour is a bit different. If there is enough fuel in the reserve to take a 2nd look for the last vehicle, the MAV should always do so without waiting for a classification from the operator.

It is difficult to observe the 2nd looks from the plots of the mission, so a printout accompanies each run and indicates for each MAV where 2nd looks were taken, and an animation replay tool was developed to observe the behavior. For example, for figure 5, every operator classification was "NOF" (this is not uncommon – with the given problem parameters, on average, one gets one, sometimes two readings of "OF" per run, with 24 total vehicles of which 6 are of interest). The red and green MAV always take 2nd looks without waiting for the operator's feedback.

veh1	veh2	veh3	veh4	veh5	veh6
25	14	21	18	14	15

Table 4. Delays (rounded) for MAV 2 (blue), in the run shown in figure 5.

MAV2, shown in blue in figure 5, had a tour that allowed it to take at least two 2nd looks. It took a second look at the second vehicle in its tour, where it drew a short delay (14), and then again at the 5th vehicle, for which the delay was also short. It then took a "default" 2nd look at the 6th vehicle, as it had enough flight time.

veh1	veh2	veh3	veh4	veh5	veh6
18	13	27	15	23	17

Table 5. Delays (rounded) for MAV 3 (yellow), in the run shown in figure 5.

MAV3, shown in yellow in figure 5, had a tour that allowed it to take at least four 2nd looks. It did not take a 2nd look at the first vehicle, because the operator was overloaded and time slipped. It did take a 2nd look at the second, fourth, fifth, and sixth vehicles.

The calculation of critical intervals can be adapted to reflect better characterizations of the delays when more information is available – in particular, a skewed distribution such as a shifted gamma distribution might more accurately represent the delays. Whatever the distribution chosen, the intervals described in section IIA can be computed (it might have to be done numerically for complicated distributions).

The case of MAV3, as discussed above, brings out an interesting coupling between tour planning and operator workload. If a greedy strategy is employed to plan the tours, each MAV will fly to the vehicle closest to it first. A consequence of this strategy is that in the first part of the mission, the operator is quickly overwhelmed by the number of requests for 2^{nd} looks, and saturates. This is alleviated somewhat by using the scheduler discussed in section IIB. In addition, whenever a MAV will fly over all vehicles in his tour regardless of operator classifications, that MAV's classifications can be regarded as non-time critical. In practice, spacing out the flyover of vehicles at least for the first few vehicles of the tours can be dealt with by the tour allocation mechanism. It is unrealistic to expect this solution to function well after a few vehicles because the 2^{nd} look decisions are not known in advance. Also, with a greedy allocation of tours to MAVs, the operator is under-utilized near the end of the mission as the vehicles tend to be further away at the end of the mission.

Another interesting feature of the simulation is that the MAV expects an answer about each vehicle before it reached the next vehicle in its tour. This is not convenient in practice, and so the framework should be changed to accommodate for clusters of relatively closely spaced vehicles. In that case, there will be significant advantage to paying the U-turn cost once, and revisiting several vehicles in one shot.

Furthermore, the simulations indicate that fairly regularly (about 5 to 15 times per average mission), the MAV will fly over vehicles that have not been selected by the SAV. As the SAV is not perfect, it might be interesting to take non-time critical pictures of those vehicles as well. In addition, paths could be planned to gather as many of these images as possible without affecting the efficiency of the rest of the mission.

Finally, there are a number of cases where the path planning yields a "free" 2^{nd} look, as all vehicles are not aligned in a straight line to start with, and fairly regularly a MAV will have to make a 180 degree turn after it has visited a vehicle to go back towards its next vehicle. Whenever this happens, a 2^{nd} look picture should be taken.

In addition to the above considerations, it is worthwhile to consider two additional special cases: corrupted images, and images with low information content (for example, low contrast). Some of these cases can be detected automatically before an image is presented to the operator.

A corrupted image may occur because of an information storage or transmission problem. A large category of corrupted images will have large blocks of corrupted pixels, for example that are all at the same value. This can be detected automatically. If an image is corrupted, it is possible to automatically request a 2^{nd} look fairly quickly (this can be done with the scheduler in conjunction with a pre-processing module). If an image has low information content, there can be several causes. Low contrast, for example, may be caused by shadows, foliage etc… This type of image problems may also be detected automatically, and may call for not taking a 2^{nd} look (the conditions will not have improved). A more tricky problem is that of images that do not contain the vehicle (for example, the MAV was blown off course by a wind gust, was out of range, or at the wrong altitude, and the picture is unusable). It may be possible to automate the detection of this type of image, and to quickly request a 2^{nd} look in this case.

OPERATOR CONFUSION

We now relax the assumption that the operator is a perfect classifier, and consider the case of an operator confusion matrix of the form:

	Op says F (OF)	Op says NF (NOF)
Feature in frame (F)	.95	.05
No feature in frame (NF)	.2	.8

Table 6. Operator confusion matrix.

One can draw the decision tree and compute the probability, for example, that a given vehicle is of interest, given an operator answer of "OF". This yields the following results for the 1st look.

P(OF) = 25% P(T|OF) = 39.24%
P(NOF) = 75% P(T|NOF) = 20.3%

The operator confusion has a strong influence on the quality of the sensor (with a perfect operator, P(T|OF) = 1). More information is now obtained from taking 2nd looks if the initial reading is "OF", to try to gather more information and confirm the 1st pass reading, so 2nd looks are always taken for classifications of "OF". For a reading of "NOF", the 2nd look decisions proceed as detailed above for the case of a perfect operator.

The information gathered at the 2nd look is as follows:

P(OF,OF) = 5% P(T|OF,OF) = 52%
P(OF,NOF) = 17% P(T|OF,NOF) = 39.5%
P(NOF,OF) = 17% P(T|NOF,OF) = 39.5%
P(NOF,NOF) = 61% P(T|NOF,NOF) = 14%

Depending on the information gathering goals of the mission, knowing that a vehicle is of interest with probably 52% (or 39.5%) may not be sufficient. Note that this is still a significant improvement over the 25% probability from the SAV. Two different directions can be explored to improve the performance of the system. One is to study the possibility of taking more than 2 looks at each site, with additional looks being correlated to the initial looks. The other is to conduct sensitivity analysis and determine which factors are most important in the performance, and focus on improving these factors if possible.

One can consider the case of 3rd looks (or more), where the 1st and 3rd looks are correlated, to improve the system information. Depending on the degree of correlation between the 1st and 3rd look, it might be worthwhile to take more passes to try and obtain more information. For completely uncorrelated passes, it is worthwhile to take a third look if the reading at the first look was "OF", to try and confirm that reading. That is still true for the correlated case, as long as the correlation is not perfect, even though the benefits are much less.

TAKING MORE THAN 2 LOOKS

For uncorrelated looks, the following decision tree applies:

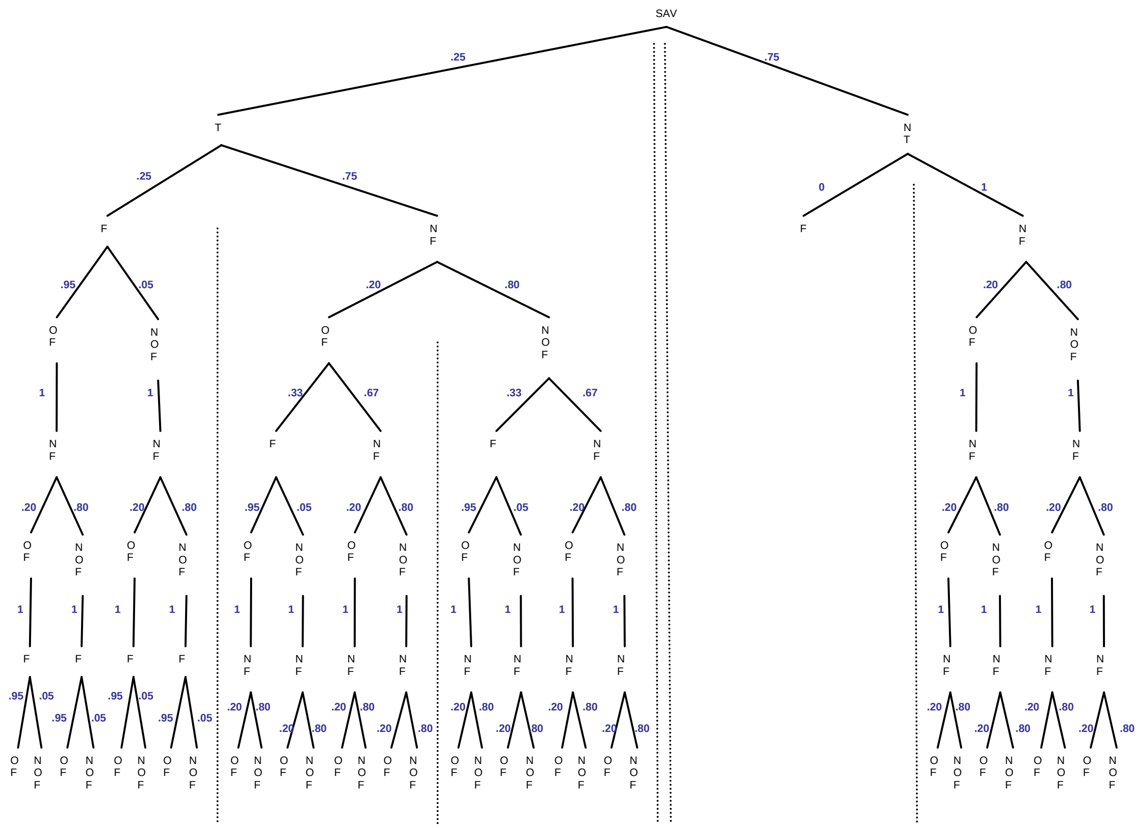

Figure 8. Decision tree for 3 looks, uncorrelated.

And we get:

P(T|OF,OF,OF) ≈ **100%**
P(T|OF,OF,NOF) = 41%
P(T|OF,NOF,OF) ≈ **100%**
P(T|OF,NOF,NOF) = 18%
P(T|NOF,OF,OF) = 41%
P(T|NOF,OF,NOF) = 39.5%
P(T|NOF,NOF,OF) = 15.5%
P(T|NOF,NOF,NOF) = 13.6%

We can compute expected values for the different tours (for now, let's use the intuitive, least squares form (this is at each site):

$$V(look) = \left[p_i(look) - \frac{1}{N}\sum_{i=1}^{N} p_i(look) \right]^2 - \left[p_i(prev\_look) - \frac{1}{N}\sum_{i=1}^{N} p_i(prev\_look) \right]^2$$

Δ_EV(1st look) $\approx$ 0

Δ_EV(2nd look) = 0.0179
  Δ_EV(2nd look|OF) = 0.038
    Δ_EV(2nd look|NOF) = 0.01234

Δ_EV(3rd look) = 0.0493
  Δ_EV(3rd look|OF,OF) = 0.087
  Δ_EV(3rd look|OF,NOF) = 0.133
  Δ_EV(3rd look|NOF,OF) = -0.0034
  Δ_EV(3rd look|NOF,NOF) = 0.038

Getting two consistent (in the right locations, that is, 1st and 3rd slots) 'OF' readings out of three readings is pretty much enough to have really high confidence that the object of interest is indeed a target. <u>This is only if the 1st and 3rd pass are not correlated.</u>

So we can get some <u>**heuristics**</u> and combine these with the critical time delay method.

Take a 2nd look most if 1st look = 'OF', but if you have time take 2nd looks also for 'NOF'
Take a 3rd look most (in order of priority)
  if 1st look was 'OF', especially if the 1st and 2nd look are 'OF,NOF'
  if the 1st and 2nd look are 'OF,OF'
  or if the 1st and 2nd looks were 'NOF,NOF'.

TAKING A LOOK AT CORRELATION BETWEEN LOOKS

Say we do take a third look. Likely the information contained in looks 1 and 3 is correlated.

Numbers quantifying this are hard to find on the net (perhaps the HE guys would know).

So, let's take some educated guesses. Use subscripts to indicate the look. The second column contains uncorrelated values.

P(OF3|OF1, F) = .99	P(OF1|F)=.95
P(NOF3|OF1, F) = .01	P(NOF1|F)=.05
P(OF3|NOF1, F) = .8	P(OF1|F)=.95
P(NOF3|NOF1, F) = .2	P(NOF1|F)=.05
P(OF3|OF1, NF) = .5	P(OF1|NF) = .2
P(NOF3|OF1, NF) = .5	P(NOF1|NF) = .8
P(OF3|NOF1, NF) = .5	P(OF1|NF) = .2
P(NOF3|NOF1, NF) = .5	P(NOF1|NF) = .8

Draw the tree, compute all values:

P(T|OF,OF,OF) = 57.26%

P(T|OF,OF,NOF) = 21.47%
P(T|OF,NOF,OF) = 57.45%
P(T|OF,NOF,NOF) = 9.64%
P(T|NOF,OF,OF) = 39.12%
P(T|NOF,OF,NOF) = 35.97%
P(T|NOF,NOF,OF) = 15.31%
P(T|NOF,NOF,NOF) = 14.86%

Δ_EV(3rd look, correlated) = 0.0092
Δ_EV(3rd look, correlated|OF,OF) = 0.0196
Δ_EV(3rd look, correlated|OF,NOF) = 0.04736
Δ_EV(3rd look, correlated|NOF,OF) = 0.00037
Δ_EV(3rd look, correlated|NOF,NOF) = 0.00012

Compare to the uncorrelated values:
Δ_EV(3rd look, uncorrelated) = 0.0493
Δ_EV(3rd look, uncorrelated|OF,OF) = 0.087
Δ_EV(3rd look, uncorrelated|OF,NOF) = 0.133
Δ_EV(3rd look, uncorrelated|NOF,OF) = -0.0034
Δ_EV(3rd look, uncorrelated|NOF,NOF) = 0.038

Taking a 3rd look is still particularly worth it if the 1st look reading is OF. Still almost not worth it at all if 1st look reading was NOF. Benefit of 3rd look is about 1/5 of uncorrelated case.

Equation used for computing value of tour: (same as previously)

$$EV(look3 \mid X,Y) = \left[\frac{\left[(P(T \mid X,Y,Z) - P(T \mid X,Y)) * P(X,Y,Z) + (P(T \mid X,Y,NZ) - P(T \mid X,Y)) * P(X,Y,NZ)\right]}{P(X,Y)}\right]^2$$

SENSITIVITY ANALYSIS

In terms of trying to understand which factors are most important to the information gathering aspects, let us try to understand what variables affect, for example, P(T|OF) = 39.24% on the first pass. Sensitivity analysis is concerned with a local measure of the effect of a given input on a given output. We started by setting distributions for the input factors. For example,

- αor (the ratio of vehicles of interest to clutter vehicles) is taken to be uniformly distributed between .05 and .15 (nominal 0.1)
- The probability of a true positive from the SAV, PTT, is taken to be uniformly distributed between .4 and .8 (nominal, 0.6)
- The probability of a false positive from the SAV, PTNT, is taken to be uniformly distributed between .1 and .3 (nominal 0.2)
- The probability of a true positive from the operator, POFF is taken to be uniformly distributed between .9 and 1 (nominal 0.95)
- The probability of a false positive from the operator, POFNF, is taken to be uniformly distributed between .1 and .3 (nominal 0.2).

We are assuming theta (field of view of feature) is fixed, as it is not likely that this will be adjustable (it is a feature of "enemy", not "friendly" vehicles).

We then estimate the first-order sensitivity indices by calculating derivatives of P(T|OF) with respect to the variables. This is according to intuition, as well as to most of the literature on sensitivity analysis [5]. The coefficients obtained are normalized by their variance divided by the total variance.

$$S_{\alpha_{OR}} = .318 \qquad S_{PTT} = .587 \qquad S_{PTNT} = -.440$$
$$S_{POFF} = .018 \qquad S_{POFNF} = -.339$$

The sum of the square of the coefficients does not total one, which indicates that there are cross-effects, that is, the combined effect of two (or more) factors is greater than the sum of individual effects. These cross-effects can be determined by computing higher-order derivatives.

However, the results above are interesting in their own right. The highest benefit comes from increasing the SAV true positive probability of detection. Significant gains can also be obtained by decreasing the probabilities of false positives from both the SAV and the operator. Finally, increasing the original ratio of vehicles of interest to clutter vehicles (picking areas where this ratio is known to be high ahead of time) also significantly helps the information gathering abilities of the system.

ENGAGING THE ENEMY:

MECHANISMS FOR INCLUSION OF THE ADVERSARY'S RESPONSE

We are considering options for including mechanisms for actions/responses of the red force given actions of the Blue force. Game theory uses mathematical models to model human decision making in competitive situations. It is ideally suited for analyzing military situations because it depicts the realistic situation in which both sides are free to choose their "best" moves and adjust their strategy over time.

The method consists of the following steps:

1. Determine the tactical options available to each side.
2. Assign a numerical value to each possible outcome.
3. Calculate all possible strategies and their outcomes.
4. Find each side's optimum strategy.
5. Determine the expected result of the game.

A POSSIBLE STRATEGY FOR RED

If you see a Blue MAV, make 1 call to the red site closest to you. This call may or may not go through. If the "closest" red site gets the call, it's occupants will camouflage their setup better (probability of saying for sure it is a target divided by two, for example). This "closest" red guy may not be the next guy in the Blue MAV's sequence. In fact, he may not even be on this Blue MAV's sequence at all. Red only calls if he sees a MAV, and then only one call each time.

Red's state space:
- position of all red sites
- camouflage state of all red sites $\in \{1 = \text{high}, 2 = \text{normal}\}$

Red's information structure:
- at each site, either Red sees a Blue MAV or he doesn't

Red's space of actions:
- If Red sees a Blue MAV at site i, it calls the nearest site to i, and improves its camouflage state with a given probability (success in being called and warned), and for some time T (can't stay camouflaged forever).

Red's strategy is represented by the mapping of his information structure onto his actions.

A POSSIBLE STRATEGY FOR BLUE: SINGLE MAV OPERATIONS

To try and minimize the effects of Red's strategy, use tour planning, plan to visit sites that are near corners of the grid first to minimize the odds that your target has been called and has camouflaged. Regardless, Red's strategy is bound to hurt Blue's results. Starting off in the corners will use more fuel and limit ability to take 2nd looks. Whether or not to use this strategy will depend on the probability that Red's call goes through. Seems like those things could be jammed pretty easily.

Note that Red's communications graph is most likely not be connected. To have a connected graph, his sites would have to be equidistant, and a direction of information defined, and we could pick probably search the map and pick them out. A situation where all red sites are roughly equidistant and Red gets two phone calls, one in each direction, would be harder.

A POSSIBLE STRATEGY FOR BLUE: COORDINATED MAV OPERATIONS

Send two vehicles in a team. Veh1 flies over the site, and vehicle 2 waits by the closest enemy site, according to us. When veh1 flies over his site, start timer, wait 1 minute, then have veh2 fly over nearest guess, and see if we can catch the Red camouflaging (guys running around). If there is camouflaging activity, then we have a target for sure. If there is no camouflaging activity, then we may be missing a target, and we know bounds for where it might be.

In fact, consider the Girard conjecture (will attempt to prove, after end of project this summer): If Red makes one call each time, Blue has an optimal strategy involving 2 vehicles. If Red makes n calls each time, Blue has an optimal strategy involving n+1 vehicles.

Note that this type of strategies is most likely unhelpful with the current vehicles, as they do not have sufficient air-to-air communication capabilities, or onboard processing power.

REVIEWING BASIC GAME THEORY: A SIMPLER PROBLEM

Consider a game between two players (red and blue) who pursue opposite goals. Red (the "attacker", that is, the MAV) must choose one of two possible sites to visit for surveillance, and Blue (the "defender") must decide how to best camouflage them.

We assume at first that Blue has a finite number of assets available for camouflage (for example, tarps). To make these assets effective, they must be assigned to a particular site, and Blue must choose how to distribute them among sites.

To raise the stakes, let's assume that each tarp only provides partial camouflage of a site (for example, masks a 10deg range of aspect angles, or divides the probability of detection by 2, or something like that), and that Blue only has three tarps available, and is faced with the decision of how to distribute them among the two sites.

We start by assuming that both players make their decisions independently and execute them without knowing the choice of the other player.

We can use the ***cost*** below, which Blue tries to minimize and Red tries to maximize:

$J = c_0$ if 0 tarps camouflage site visited
$J = c_1$ if 1 tarp camouflages site visited
$J = c_2$ if 2 tarps camouflage site visited
$J = c_3$ if 3 tarps camouflage site visited

Implicit in this is the notion that both sites have the same strategic value. (This may not be true). Without loss of generality we can normalize these constants to have $c_0 = 0$ and $c_3 = 1$. We consider arbitrary values for c1 and c2, with the (reasonable) constraint that $0<c_1\leq c_2<1$.

As formulated above, Red has to possible choices (visit site 1 or site 2), and Blue has a total of four different ways of distributing its tarps among the two sites. Each choice available to a player is called a ***pure policy*** for that player.

We will denote by ui, $i \in \{1,2\}$ and vj, $j \in \{1,2,3,4\}$ the policies available to Blue and Red respectively. These policies are enumerated in the tables below.

Blue's policies:

Policy	Site assigned
u_1	1
u_2	2

Red's policies (each x denotes a tarp):

Policy	Site 1	Site 2
v_1	xxx	
v_2		xxx
v_3	xx	x
v_4	x	xx

Red's v_1 and v_2 policies are called 3-0 configurations and the policies v_3 and v_4 are called 2-1 configurations.

The game can be represented in its ***extensive form*** by associating each policy of Blue and Red with a row and a column, respectively, of a matrix G. The entry g_{ij}, $i \in \{1,2\}$ and $j \in \{1,2,3,4\}$ of G corresponds to the cost J when Blue chooses policy u_i and Red chooses policy v_j. For his game, G is given by:

$$\begin{array}{cc} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ G \equiv & \begin{bmatrix} 1 & 0 & c_2 & c_1 \\ 0 & 1 & c_1 & c_2 \end{bmatrix} \begin{array}{c} u_1 \\ u_2 \end{array} \end{array}$$

In the context of ***non-cooperative zero-sum games***, such as the one above, optimality is usually defined in terms of a ***saddle-point or Nash equilibrium***. A Nash equilibrium in pure policies would be a pair of policies {u_i*, v_j*}, one for each player, for which:

$$g_{i^*j} \le g_{i^*j^*} \le g_{ij^*} \quad \forall i, j$$

Nash policies are chosen by rational players since they guarantee a cost no worse than $g_{i^*j^*}$ for each player, ***no matter what the other player decides to do***. As consequence, playing at a Nash equilibrium is "safe" even if the other player discovers our strategy of choice. It is also a reasonable choice as the player never does better by deviating unilaterally from the equilibrium.

Not surprisingly, there are no Nash equilibria in pure policies for the game described above. In fact, all the pure policies violate the "safe" condition, that is, suppose that Blue plays policy u_1. This choice is not safe in the sense that if Red guesses the choice, he can play strategy v_1 and subject Blue to the highest possible cost. Similarly, u_2 is not safe either and cannot be in a Nash equilibrium pair.

To obtain a Nash equilibrium, one needs to enlarge the policy space by ***allowing each player to randomize among its available pure policies***. In particular, suppose blue chooses policy u_i with probability b_i and Red chooses policy v_j with probability r_j. If the game were played repeatedly, the expected value of cost is given by:

$$E[J] = \sum_{i,j} b_i g_{ij} r_j = b'Gr$$

Let's call the set of all vectors $x = \{x_i\} \in \Re^n$ for which $x_i \ge 0$ and $\sum_i x_i = 1$ the n-dimensional simplex.

Each vector b={b_i} in the 2-dimensional simplex is called a mixed policy for Blue, and each vector r={r_j} in the 4-th dimensional simplex is called a mixed policy for Red.

One of the main results in game theory, the minimax theorem, states that ***at least one Nash equilibrium in mixed policies always exists for finite matrix games***.

In particular, there always exists a pair of mixed policies, {b*, r*}, for which:

$$b'^*Gr \le b'^*Gr^* \le b'Gr^* \quad \forall b, r$$

Assuming that both players play at the Nash equilibrium, the cost will then be equal to b'*Gr*, which is called the value of the game.

It is straightforward to show that the unique Nash equilibrium for the game considered above is given by:

$$b^* = \begin{bmatrix} 1/2 & 1/2 \end{bmatrix}^T$$

$$r^* = \begin{cases} \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \end{bmatrix}^T & c_1 + c_2 \leq 1 \\ \begin{bmatrix} 0 & 0 & 1/2 & 1/2 \end{bmatrix}^T & c_1 + c_2 > 1 \end{cases}$$

with value equal to:

$$b'^*Gr = \max\left\{\frac{c_1 + c_2}{2}, \frac{1}{2}\right\}$$

The equilibrium corresponds to the intuitive solution that Blue should randomize between visiting site 1 or site 2 with equal probability, and red should randomize between placing most of its tarps near site 1 or site 2 with equal probability. The optimal choice between 3-0 or 2-1 configurations depends on the parameters c_1 and c_2. The 3-0 configurations are optimal when $c_1+c_2 \leq 1$, otherwise 2-1 configurations are optimal.

THE GORDIAN KNOT TO AUTO-FLY

The problem as defined above is a bit broad. So we might start by making a few assumptions:

Assumption 1: "Auto-fly", not "cooperative auto-fly". We will not consider problems relating to the cooperative operations of multiple UAVs.

Assumption 2: This is a technical white paper. We are assuming the "Gordian knot" technology is "free". We will not consider costs.

Assumption 3: Security, in the sense of secure communications, not being able to be jammed or listened in on, or having control of the UAV stolen by the enemy, etc…, is assumed solved.

Fact: There is a large range of UAV platforms and operations. Different technologies will be useful for different scale and types of vehicles.

Broad types of missions:

Type 1: emphasizes autonomy, survivability and weapons (combat, fighter/bomber)
Type 2: emphasizes payload capacity and persistence (reconnaissance)

Type 1 missions might include radar jamming and destruction, SEAD, and weapon delivery (with varying levels of autonomy). Integration with manned aircraft will be a milestone.

Type 2 missions might include persistent ISR, establishing communication relays, patrolling, aerial refueling, and maybe airlift. Payload power and weight are a big issue, as is endurance (> 24 hrs).

Broad types of vehicles:

MAV, SAV, UAV

MAV: up to 1ft in wingspan
Examples: batcam, Stanford helicopter (Ilan Kroo)

SAV: up to roughly 10ft in wingspan
Examples: Dragoneye, MLB bat, ACR silver fox

UAV: 10-150ft in wingspan
Examples: UCAV, Predator, GlobalHawk

Broad types of technologies:

Platforms:
- endurance
- signature
- propulsion (especially at small scales)
- survivability (tactics, technology and cost)

Payloads:
- resolution
- power
- weight

Communications:
- data rates
- standards

Computing, controls, operators
- autonomy
- standards and interoperability
- strategies for wind (especially at small scales)
- see and avoid

WHAT TO CHOOSE?

Here's a list of "Gordian knots" that come to mind, roughly in the order they would be chosen by the authors.

1. Technologies (perhaps computer-aided systems) to overcome psychological reluctance to transition to radically new technologies/capabilities, and to overcome policy barriers.
    a. FAA
        i. See and avoid
        ii. Collision warning
        iii. Lost link procedures
        iv. Mishap rates
        v. All weather practices
        vi. Instrument Flight Rules etc…
    b. Access to airports/airspace
        i. Including in foreign countries
    c. Passenger willingness to fly on a plane with no crew

   d. Pilot willingness to fly alongside unmanned vehicles
   e. Commander willingness to trust unmanned vehicles are autonomously doing the right thing

2. Standards for forward/backward compatibility of vehicles and/or systems, and interoperability
   a. UAV to operator
   b. UAV to UAV
   c. UAV to manned system
   d. UAV to other unmanned vehicles (for example UGV)
   e. Communications and messaging standards

3. Damage assessment of self and other unmanned vehicles
   a. Look at the space shuttle fiasco…
   b. Also, have capability to inspect other vehicles, and make assessment (e.g., your tail is half gone)

4. Continuous adaptation to instance of mission, conditions etc…
   a. Logging data to improve performance (experimental data or simulation)
   b. In a system with a large number of tunable parameters (say 100), how to continuously adapt to account for the conditions on a given day? (For example, in abstracted counter scenario, might have particularly good intelligence on clutter/target ratio on a given day, or, might want to retune a PID loop, etc… This is too technical to expect the operator to do it, so it should be done automatically. The operator should be able to choose between maybe 5 configurations).

5. Ability to safely transfer control authority between different control centers, for example various human operators, including soldiers on the ground for close air support.

6. Accounting for human, including the operator and the enemy.
   a. How to define default behaviors, how to adjust to the context of an operation
   b. Plan on several time scales (seconds versus minutes), and sound alarm if operator input is late/missing.

CONCLUSIONS

One fundamental issue that was not formally discussed yet in the report is the problem of the value of the information gathered by the MAV/system over the course of the mission. The goal is to gather the most information about the state of the vehicles in our world, particularly those elected by the SAV. We can use a measure of the value of information that is based on getting large variations in the probabilities that a vehicle is of interest or clutter, as compared to the default value.

$$V(look) = \left[p_i(look) - \frac{1}{N}\sum_{i=1}^{N} p_i(look) \right]^2 - \left[p_i(prev\_look) - \frac{1}{N}\sum_{i=1}^{N} p_i(prev\_look) \right]^2$$

There are many other techniques to assign value to information, starting with the work of Shannon and its many variations. For example, it might be possible to use a normalized form of Shannon entropy, such as:

$$V = 1 + \frac{\sum_{i=1}^{N} p_i \log(p_i)}{\log(N)}$$

However, we find this form to be less intuitive to use.

Another possibility, closer to higher-level decision making, is to assign "x points" to vehicles of interest identified correctly, and "-y points" to false positives, and aim to maximize the number of "points". This involves some fundamental trade-offs about the value of real targets (vehicles of interest) versus the cost of collateral damage (false positives). In this type of scenario, one might want higher probabilities that a vehicle is indeed of interest. How good the sensor needs to be in this scenario is yet to be established. Finally, cost benefit analysis will have to be conducted. Given the value of the information collected by the MAV, is the cost acceptable?

Our analysis above suggests that small improvements in the confusion matrices of both the SAV and the operator will yield big improvements in the quality of the information collected.

Finally, the analysis is preliminary in terms of human effectiveness engineering, and much work has yet to be done. Final number and better characterizations of the human operator will be obtained from field test data. One consideration to remember is that the system is built to be optimal stochastically, in the long term. It may not yield an optimal answer to any given run, or mission. This may cause some frustration in the operator, and the operators should be briefed early and often on how the decision making works and what the effects may be ahead of time to alleviate this problem.

REFERENCES

[1] C. Derman, G.J. Lieberman and S.M. Ross, "A Sequential Stochastic Assignment Problem", in Management Science, vol. 18, no. 7, March 1972, pp. 349-355.

[2] M. L. Puterman, "Markov Decision Processes: Discrete Stochastic Dynamic Programming", Wiley Series in Probability and Statistics, Wiley, 2005.

[3] S.M. Ross, "Introduction to Stochastic Dynamic Programming", Academic Press, 1983.

[4] A. Silberschatz, P. B. Galvin and G. Gagne, "Operating System Concepts", John Wiley and Sons, 2004 (7th Edition).

[5] A. Saltelli, S. Tarantola, F. Campolongo and M. Ratto, "Sensitivity Analysis in Practice", Wiley, 2004

APPENDIX A : TENTATIVE SUMMER SCHEDULE (JUNE 2005)

Short problem description

A MAV (Micro Air Vehicle) has to fly over N sites for classification purposes. The list of sites is provided by a planner and the sequence of sites is fixed. A certain fraction of the sites is known to be targets. The MAV flies over each site, takes a reading (for example, a picture) and transmits the reading to a human operator for target recognition. The MAV flies towards its next target as it waits for an answer from the human operator.

After some delay, the operator answers with a classification of either "type A" or "type B". "A" indicates that the object is a target with probability $p(T|A) = 1$. "B" indicates that the site is a target with probability $p(T|B)$, such that $0<p(T|B)<1$, that is, "B" indicates some ambiguity about the site.

When the answer from the operator is received, the MAV has the option to either continue on to the next target, or turn around and go take a 2nd look at the site. If the MAV takes a 2nd look, he will get another reading (either "BA", target, or "BB", still ambiguous). The cost of taking a second look includes a fixed cost to turn around (the cost of changing direction by 180 degrees, twice), plus the delay caused by having to travel back to the first target again, and back. The MAV has limited flight time, M.

We know the following probabilities about the problem: $p(A)$, $p(B)$, $p(T|A)$, $p(T|B)$, $p(T|BA)$, $p(T|BB)$. No further information is gained by taking more than 2 readings.

Level 0: Derive an optimal policy that chooses between possible control actions (continue, 2nd look), given statistics about the target distribution, the result of the classification from the 1st look, transmission and operator delays, and cost to take a second look.

Level 1: Include a more complete **model of the operator**, including a better description of the system delays, a characterization of operator workload, an operator confusion matrix, and the possibility of image degradation.

Level 2: Include a **characterization of the adversary**, and of his possible response to the MAV searching.

Level 3: Consider possible **coupling** of the MAV behaviors/trajectories.

A more precise description of the scenario, particularly the target characterization, is given in [Chandler, Pachter].

Schedule

Week 1: Problem Formulation
Phrase problem as a DP problem
Consider different objective functions

Week 2: Solution to basic level 0 problem using stochastic sequential assignment following [Derman et al]. The method allows for the computation of critical thresholds above or below which actions should be taken. These critical thresholds depend on the number of sites and on the cumulative distribution function of the delays. The cost function r(p,x) must be differentiable and satisfy the following criterion, where p indicates whether a 2nd look should be taken, and x is the random delay.

$$\frac{\partial}{\partial x}\frac{\partial}{\partial p}r(p,x) \geq 0$$

Compute critical thresholds for abstracted COUNTER scenario using uniform probability distribution (the delay is some random number between 0 and 9). The cost is considered to be the time spent to turn back and take a second look. The expected benefit of taking a second look is the same at each decision point.

Weeks 3, 4, 5 (expected): Consider more realistic operator model.

Weeks 6, 7, 8 (expected): Consider adversary reactions.

References:

P. Chandler, “Abstracted Counter Scenario”
M. Pachter, “ATR Module Modeling”
C. Derman, G.J. Lieberman and S.M. Ross, “A Sequential Stochastic Assignment Problem”, Management Science, Volume 18 Number 7, March 1972, pp 349-355

Notes and comments on tentative summer schedule:

A simulation effort to validate results was undertaken that hadn’t been budgeted for in original schedule. Setting up and debugging the simulation took a little while, but critical insights were gained that really improved the quality of the decision making strategies, and the overall results. Also, hopefully this has made the resulting product more useful for an actual implementation. And the code is available to test interactions between modules etc. All code was written in Matlab for easy interfacing to MultiUAV. However, it delayed the consideration of adversary reactions by about 2 weeks.

Level 1: Include a more complete **model of the operator**, including a better description of the system delays, a characterization of operator workload, an operator confusion matrix, and the possibility of image degradation: **completed.**

Level 2: Include a **characterization of the adversary**, and of his possible response to the MAV searching: **partially completed**. Characterization itself is complete, yet properties of good responses from the Blue force not proven.

Level 3: Consider possible **coupling** of the MAV behaviors/trajectories: **partially completed**. Couplings were discussed and identified. Not fully complete as tour/path planning strategies can be considered as a separate problem.

In addition, still looking at Nash equilibrium formulation for switching vehicles between teams.